

20 LECCIONES DE BASIC
PARA APRENDER CON EL SPECTRUM

INGELEK



JACKSON

- Inconvenientes del ordenador
- Mantenimiento: teclado, unidad central, televisor, memoria de masa, impresora
- Intérpretes y compiladores
- Funciones numéricas
- EXP, LN, DEF FN
- Usar y definir las funciones
- Videojuego N.º 16

16

Spectrum

16K/48K/PLUS

VIDEO BASIC
Una publicación de
INGELEK JACKSON
Director editor por INGELEK:
Antonio M. Ferrer
Director editor por JACKSON HISPANIA:
Lorenzo Bertagnolio
Director de producción:
Vicente Robles
Autor: Softidea
Redacción software italiano:
Francesço Franceschini,
Stefano Cremonesi
Redacción software castellano:
Fernando López, Antonio Carvajal,
Alberto Caffarato, Pilar Manzanera
Diseño gráfico:
Studio Nuovaidea
Ilustraciones:
Cinzia Ferrari, Silvano Scolari,
Equipo Galata
Ediciones INGELEK, S. A.
Dirección, redacción y administración,
números atrasados y suscripciones:
Avda. Alfonso XIII, 141
28016 Madrid. Tel. 2505820
Fotocomposición: Espacio y Punto, S. A.
Imprime: Gráficas Reunidas, S. A.

ISBN del tomo 4: 84-85831-20-9
ISBN del fascículo: 84-85831-11-X
ISBN de la obra completa: 84-85831-10-1
Depósito Legal: M-15076-1985
Plan general de la obra:
20 fascículos y 20 casetes, de aparición quincenal,
coleccionables en 5 estuches.
Distribución en España:
COEDIS, S. A.
Valencia, 245. 08007 Barcelona.
INGELEK JACKSON garantiza la publicación de todos
los fascículos y casetes que componen esta obra y el
suministro de cualquier número atrasado o estuche
mientras dure la publicación y hasta un año después de
terminada.
El editor se reserva el derecho de modificar
el precio de venta del fascículo,
en el transcurso de la obra, si las circunstancias del
mercado así lo exigen.
Octubre, 1985.
Impreso en España.

INGELEK



JACKSON

SUMARIO

Introducción

Después de horas y horas pasadas en soledad junto a nuestro ordenador, puede ocurrir que nos olvidemos de que también él, en el fondo,... es humano. Como todos los hombres tiene dolores y achaques variados. ¿Sus enfermedades? Si hacemos la necesaria transposición hombre-máquina, la artrosis (teclas), miopía (pantalla), sordera (grabadora, microdrive), taquicardia (chip). ¿La curación? Una sola: medicina preventiva.
Algunas preciosas sugerencias sobre cómo hay que tratarlo, pueden mejorar, y mucho, su calidad de vida. En resumen, de vez en cuando le vendrá bien un chequeo...

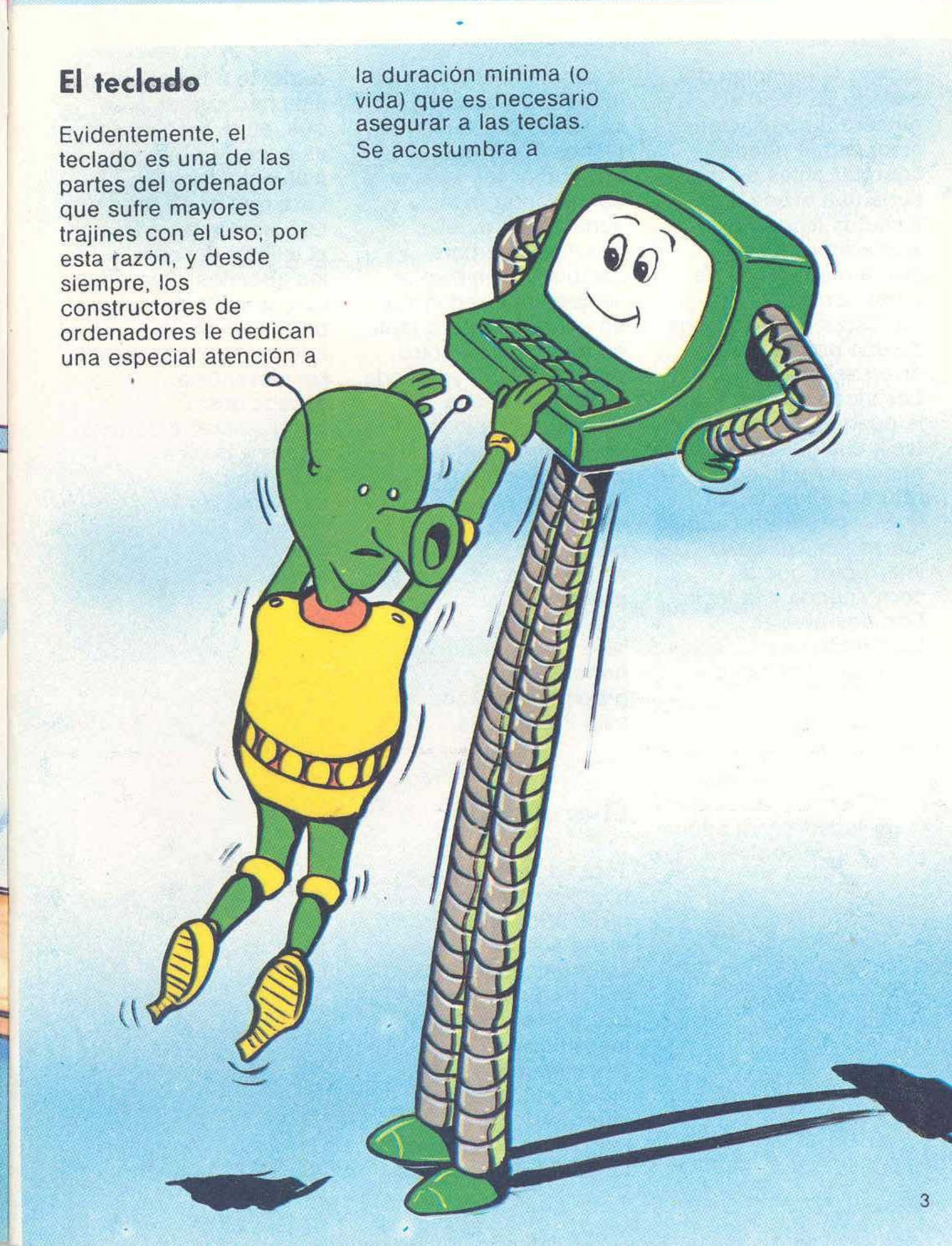
Inconvenientes y mantenimiento

De unos años a esta parte, los microprocesadores y los ordenadores personales resultan extremadamente fiables, sobre todo gracias a la constante reducción del número de componentes electrónicos que los constituyen.
La fiabilidad de un dispositivo electrónico (y un ordenador no escapa ciertamente a esta regla) es de hecho mayor cuanto menor sea la cantidad de piezas que lo componen. En comparación con sus predecesores, los ordenadores modernos son menos susceptibles de averías y más resistentes.
Sin embargo, como por otra parte ocurre con cualquier máquina construida por el hombre, también los ordenadores, (y ciertos periféricos) pueden en ocasiones estropearse y funcionar mal, perjudicando así la fiabilidad de todo el sistema.
En general, las averías más comunes se localizan en las partes del ordenador sometidas a desgaste mecánico; por ejemplo, teclado, enchufes de alimentación o de conexión, impresora y lector de discos.
Estos inconvenientes (excepto en casos particulares y extraordinarios) forman parte del ciclo manual de vida de la máquina: igual que en un automóvil se puede necesitar de vez en cuando un ajuste de neumáticos o de frenos, también es completamente normal que un ordenador necesite algún mantenimiento.
Por lo tanto, hoy intentaremos, a través de un análisis de las distintas partes que componen el ordenador, examinar las más frecuentes y principales causas de averías, sugiriendo en cada caso —y siempre que sea posible— las soluciones más adecuadas y correctas que convendrá aplicar para eliminarlas.
La primera y más importante recomendación es leer en primer lugar, y prestando la máxima atención posible, los manuales de uso de todos los dispositivos conectados al ordenador; es más frecuente cometer errores por el escaso conocimiento sobre un determinado periférico.
Además, en todos los manuales se proporcionan consejos e informaciones preciosas sobre los procedimientos de conexión, manejo y uso de las diversas unidades. Los pocos minutos dedicados a la lectura de estos manuales pueden evitar varios días de espera (además de gastos monetarios) para las posibles reparaciones debidas a no haber respetado los procedimientos correctos de operación.

Un teclado colocado demasiado arriba o demasiado abajo obliga al operador a mantener una postura incorrecta y antinatural.

El teclado

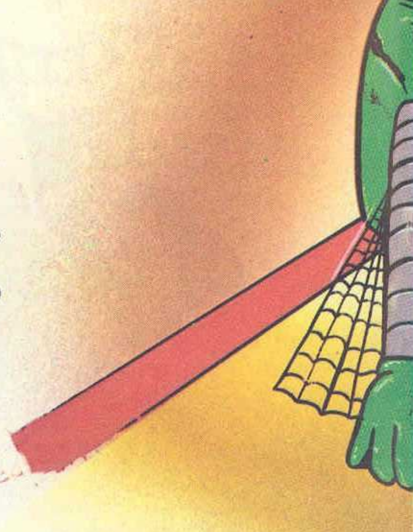
Evidentemente, el teclado es una de las partes del ordenador que sufre mayores trajines con el uso; por esta razón, y desde siempre, los constructores de ordenadores le dedican una especial atención a la duración mínima (o vida) que es necesario asegurar a las teclas. Se acostumbra a



indicar la duración del teclado en base al número de pulsaciones medias que puede soportar antes de que surja una avería. Los teclados modernos aseguran una vida media de millones de pulsaciones, correspondiente a años de uso normal de las diversas teclas.
Los inconvenientes que le pueden ocurrir a una tecla cualquiera son principalmente dos: rotura o aflojado del muelle de retorno, o mal funcionamiento del interruptor que le corresponda a la tecla. Son dos averías fácilmente identificables y que se eliminan con la simple sustitución de la pieza en cuestión.
La rotura del muelle es menos grave que la del interruptor (es suficiente con levantar la tecla y substituir el muelle, mientras que para cambiar el interruptor se tiene que recurrir a un soldador), pero tanto en uno como en otro caso, la avería se puede reparar con toda facilidad y rapidez.
Un enemigo del teclado (y, en general, de todos los elementos electrónicos) es el polvo, que, acumulándose, puede provocar falsos contactos; por esta razón, es una buena norma tapar el ordenador cuando no está funcionando.

El ordenador

Todos los elementos que componen el ordenador propiamente dicho son dispositivos electrónicos «de estado sólido». Esto significa que no son partes mecánicas o electromecánicas en movimiento, como el interruptor de encendido y apagado. La consecuencia más evidente e inmediata de este hecho radica en que, en la práctica, no es necesario ningún mantenimiento.
Para que nadie tenga que «meter la nariz» en el interior del ordenador, los circuitos integrados que componen en gran parte el aparato, pueden funcionar (salvo las inevitables excepciones) tranquilamente durante meses y meses.
Los principales enemigos de los circuitos son fácilmente identificables: polvo, golpes y líquidos. Por lo



que respecta al polvo es util lo que ya hemos dicho antes: cubrir el ordenador cuando no se usa; el polvo podria, además de favorecer falsos contactos, impedir el intercambio termico normal entre el ambiente exterior y el ordenador, provocando recalentamientos (con las consiguientes quemaduras) en los circuitos.
Hay que prestar una atención especial a posibles golpes y vibraciones. Un ordenador bien hecho está bastante protegido en este sentido y tolera sin problemas golpes y transportes, incluyendo las vibraciones del maletero de un coche; es necesario, en cualquier caso, no exagerar con los trajines ya que, a largo plazo, podrian dar problemas, debidos tal vez al desplazamiento de algun tornillo o de algun circuito interior.
Por su parte, los liquidos merecen un comentario aparte. Un liquido vertido sobre un circuito eléctrico es siempre fatal; es necesario, por tanto, acordarse de no apoyar vasos o botellas sobre ningún dispositivo electrónico. El peligro es además doble: en primer lugar, cualquier liquido caido accidentalmente provocará un cortocircuito en el interior del ordenador, con consecuencias imprevisibles pero, seguramente, desastrosas. En segundo lugar, puede acarrear un riesgo para la seguridad de las personas que se encuentren en las inmediaciones. ¡Es mucho mejor beber en otra habitación!

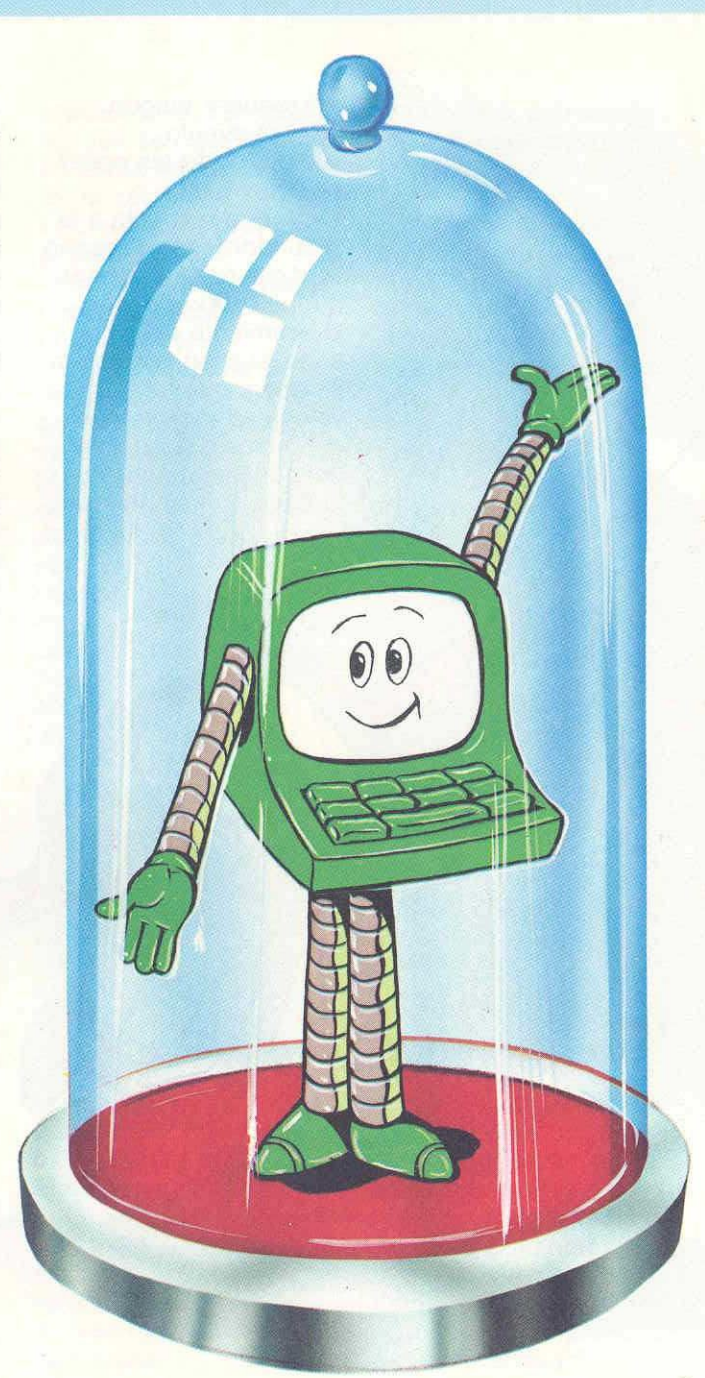
El televisor o el monitor

El mantenimiento necesario para conservar en el tiempo la eficiencia y la calidad de un televisor o un monitor es realmente mínimo, y puede ser resumido en pocas palabras: mantenerlo limpio y correctamente regulado.
Sin embargo, incluso en el caso de la pantalla se pueden recomendar algunas precauciones elementales.
Primero: evitar –como de costumbre– la acumulación de polvo.
Segundo: colocar la unidad de tal manera que el aire pueda renovarse (esta regla es fundamental). Tercero: no dejar mucho tiempo la misma imagen en la pantalla; puede suceder que los fósforos utilizados aclaren su color, imprimiendo siempre la misma imagen en la pantalla. Este inconveniente, en cualquier caso, ya ha sido limitado en los aparatos modernos.
También para la unidad de pantalla –en caso de mal funcionamiento– vale aquello que dijimos para el ordenador en sí: no tratar nunca de hacer reparaciones sin poseer los adecuados conocimientos teóricos y prácticos. Es fácil que un inexperto empeore la situación en lugar de mejorarla. Por otra parte –incluso con el cable de alimentación lejos del enchufe– en el interior del monitor o del televisor existen partes permanentemente expuestas a una elevada tensión y el contacto puede resultar peligroso para las personas. Lo único que se puede hacer es comprobar un fusible que a menudo ha sido colocado para proteger la unidad y –en el caso de que haya sido dañado– sustituirlo.

El microdrive y la grabadora

Probablemente, los peores enemigos de los dispositivos magnéticos son los malos tratos y las distracciones; casi todos los problemas que acusan pueden ser generalmente atribuidos a estas dos causas. Así, es necesario efectuar un mantenimiento periódico de las diversas partes que durante el funcionamiento de la

unidad son más utilizadas y, por lo tanto, dañadas.
En lo que respecta al microdrive, las noticias son buenas: excluyendo roturas y defectos de fabricación (que son por otra parte muy raros), la calidad de los componentes utilizados asegura un perfecto funcionamiento de la unidad por largo tiempo





sin requerir ningún mantenimiento.
Por este lado las cosas van muy bien.
En lo que respecta a la grabadora, es necesario verificar periódicamente que el calibrado y el alineamiento del cabezal sean correctos, es decir, que la velocidad de rotación del motor y la posición del cabezal respecto a la cinta contenida en el casete sean idénticas a las establecidas por el fabricante.
Para esta operación se puede recurrir a un especialista, pero se puede hacer también por cuenta propia.
Existen en el mercado cintas mediante las cuales se puede comprobar automáticamente tanto el estado del cabezal como la velocidad del motor que pone en movimiento el casete.
Naturalmente, cuando

se descubra alguna irregularidad es conveniente acudir al especialista para que ponga cada cosa en su sitio.
El desalineamiento del cabezal es un inconveniente especialmente traicionero.
Usando la grabadora con el cabezal mal alineado, generalmente todo parece funcionar de la mejor de las formas: la grabación y la carga de los casetes ocurren normalmente, sin ningún error o anomalía. Pero si se intenta cargar una cinta grabada con otro aparato (o, viceversa, se intenta cargar en otra unidad un casete grabado en ésta) la operación resultará imposible.
Este hecho deriva directamente de la posición del cabezal, que indica al lector la presencia de grabaciones distintas de las que se esperaban.
En cierta forma, el lector reconoce solamente las propias grabaciones, «personalizadas», por el desalineamiento del propio cabezal.
Generalmente, es preciso verificar la grabadora una o dos veces al año.
Más frecuente aún deber ser la limpieza del cabezal. Este debe ser limpiado regularmente para eliminar la presencia de polvo o de partículas de óxido depositadas sobre él y que entran en contacto con la cinta. La frecuencia de las limpiezas necesarias depende del ambiente (con más o menos polvo) en el que funcione la grabadora y de la intensidad con la que sea utilizada. Para este fin existen casetes especiales, ideadas específicamente para facilitar la operación.
Como regla general, el cabezal de la grabadora debe ser limpiada por lo menos una vez cada tres meses.
La velocidad y la sencillez de esta operación aconseja incluso una frecuencia aún mayor.
Finalmente, conviene recordar que también los lectores de cinta magnética son mecanismos sensibles; cuando se coloca un microdrive o una grabadora se deben también –como de costumbre– evitar golpes y vibraciones.
Un microdrive o una grabadora bien cuidados pueden funcionar durante años sin ningún problema siempre que se hagan las cosas con un mínimo de atención y –en el caso de la grabadora– de mantenimiento.

La impresora

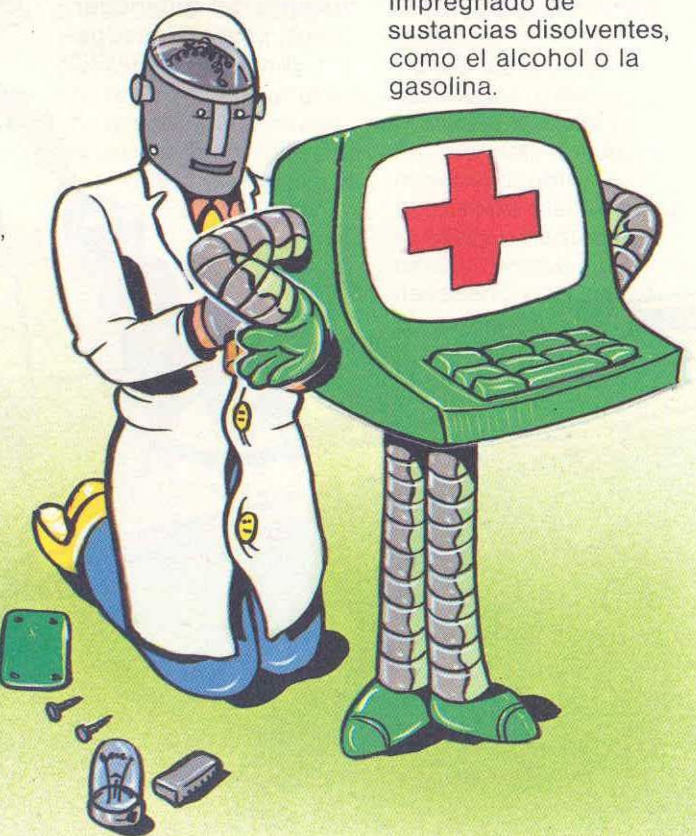
En casi todos los sistemas, la impresora es el dispositivo con el mayor número de partes mecánicas; precisamente por eso, es el más expuesto a averías, sobre todo si se emplea de modo anómalo o incorrecto.
Como una impresora tiene más partes móviles, se resiente más de un uso violento.
Veamos como hay que comportarse.
Ante todo, la impresora debe ser apoyada sobre

un soporte estable, para evitar vibraciones que la molesten en su trabajo.
Delante y detrás de la impresora debe haber un espacio suficiente para que el papel pueda entrar y salir.
El papel se inserta correctamente en el mecanismo de tracción. Esto puede parecer una perogrullada pero es fácil encajar el papel torcido, provocando así el bloqueo de la impresora. Es necesario, por tanto, aprender a respetar siempre la colocación del papel.
Un bloqueo de la impresora es debido casi siempre a una obstrucción física: a menudo el papel se bloquea porque los controles han sido regulados mal, porque ha sido introducido de forma incorrecta, o porque se han utilizado hojas inadecuadas.
Cuando el papel se bloquea, el motor de la impresora puede quemarse: es bueno, por lo tanto, no ausentarse jamás durante el funcionamiento.
También la cinta entintada (en las impresoras de agujas y en las de margarita) debe estar siempre bajo control; es necesario saber cómo insertarla correctamente, evitando los pliegues que se pueden formar durante el montaje.
El cabezal de la impresora es una de la partes más sometidas a desgastes, hasta el punto de que es casi normal que en un momento determinado se rompa. En este caso, lo único que se puede hacer es cambiarla.
En el caso de avería parcial o total de la impresora es conveniente realizar, en orden, estas operaciones:
– comprobar si la posición y la inserción de los cables de alimentación y de conexión con el ordenador son correctas;
– comprobar todos los interruptores y teclas de control;
– comprobar el estado de fusibles;
– comprobar el macanismo de impresión, la cinta y el papel;
– llegados a este momento, si las cosas todavia no funcionan, hay que llevar la impresora a un punto de asistencia técnica.
Jamás se debe usar aceite u otros lubrificantes, a no ser que esté expresamente recomendado en el manual de instrucciones.
Como siempre, las indicaciones del manual deben ser leídas y seguidas con atención.
Recuerda: cualquier impresora funcionará correctamente si está adecuadamente regulada y si se efectuan con regularidad las específicas operaciones de mantenimiento que la unidad requiere.

Los cables de conexión

A los cables de conexión les dedicaremos un cuidado y una atención especial, ya que son fundamentales para el funcionamiento de todo el sistema.
Ya que su punto débil suele ser aquél en el que el cable se inserta

en un conector, se deben observar muy cuidadosamente algunas sencillas operaciones elementales, fácilmente deducibles con un mínimo de sentido común:
– evitar siempre y en cualquier caso extraer el conector tirando del cable; esta operación resulta muchas veces fatal para la continuidad eléctrica;
– tratar, en el límite de lo posible, de que los cables no hagan pliegues bruscos;
– limitar al máximo el número de veces que se enchufan y se desenchufan los cables, puesto que con el tiempo esta operación tiende a empeorar el contacto eléctrico. Siempre es útil restablecer de vez en cuando el mejor contacto posible actuando con delicadeza sobre el enchufe mediante una pequeña pinza;
– en caso de avería, la reparación del cable es muy fácil; basta con localizar el punto de rotura (generalmente ceden siempre las soldaduras) y arreglarlo con un pequeño soldador. Es mejor evitar todas las reparaciones distintas de la soldadura (del tipo de uniones con cinta adhesiva, tornillos, cables añadidos, etc.)
– comprobar que el contacto tenga siempre lugar en las mejores condiciones, eliminando las eventuales trazas de óxido o de suciedad con un trapo impregnado de sustancias disolventes, como el alcohol o la gasolina.

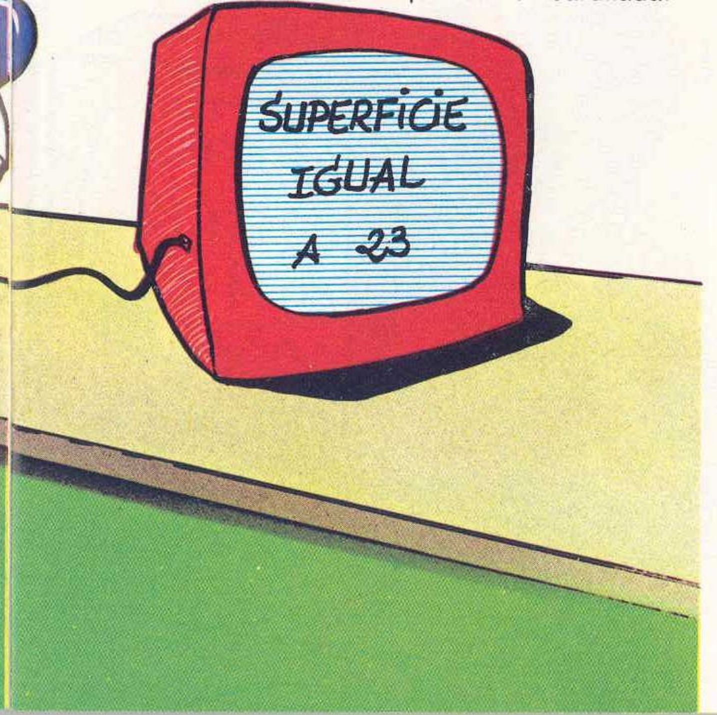




Intérpretes y compiladores

Cuando se escribe un programa en BASIC es fácil olvidar que, en realidad, nuestro ordenador no es capaz de comprender los comandos que nosotros le enviamos, y que solamente puede ejecutar las instrucciones pedidas a través de una secuencia de números binarios. Estos números, que constituyen la lengua materna del ordenador, le son proporcionados por el intérprete BASIC, que durante la ejecución traduce una tras otra las diversas instrucciones que componen el programa. El intérprete BASIC se comporta con el ordenador exactamente igual que un traductor simultáneo, que traduce instantáneamente lo que se dice a frases

comprensibles y ejecutables por la CPU.
En la práctica podemos, por tanto, imaginar al intérprete como una especie de caja mágica a la cual se proporcionan como entrada instrucciones BASIC y que responde en salida con órdenes «digeribles» para la unidad central.
En realidad, también el intérprete BASIC es un programa, cuya misión es la de traducir las instrucciones BASIC a instrucciones en código máquina.
Por muy veloz y optimizado que esté, el programa traductor precisa de un cierto intervalo de tiempo para realizar su propio trabajo: en otras palabras, la traducción de instrucciones BASIC a instrucciones código máquina necesita un cierto trabajo, y, en consecuencia, produce un determinado retraso en la ejecución.
La cuestión puede parecer embarullada: ¿por qué recurrir al traductor cuando se podría escribir el programa directamente en instrucciones ejecutables por el microprocesador, obteniendo además las respuestas mucho más deprisa? La respuesta se basa en el enorme esfuerzo que requiere —en comparación con la programación en BASIC— la escritura de programas en código máquina.
Desde luego, los resultados y las respuestas resultan mucho más rápidos e inmediatos en assembler (así se llama el código máquina), pero el tiempo necesario para escribir el mismo programa en assembler y en BASIC es mucho más favorable al BASIC (la diferencia media es de 1 a 20).
El intérprete es, por tanto, el nexo entre el hombre y la máquina, es decir, el elemento que permite que tenga lugar una recíproca y perfecta comprensión.
Pero, como hemos dicho antes, ya que el intérprete debe leer cada vez las diversas instrucciones, analizarlas, verificar la corrección de su



sintaxis y ejecutar las operaciones solicitadas, la velocidad de trabajo no puede sino resentirse, resultando notablemente inferior a la que el programa hubiera tenido si hubiese sido escrito directamente en código máquina.

De estas consideraciones se deduce que, aunque siempre sea muy superior a la de cualquier ser humano, la velocidad de ejecución de un programa BASIC no podrá superar nunca un cierto nivel.

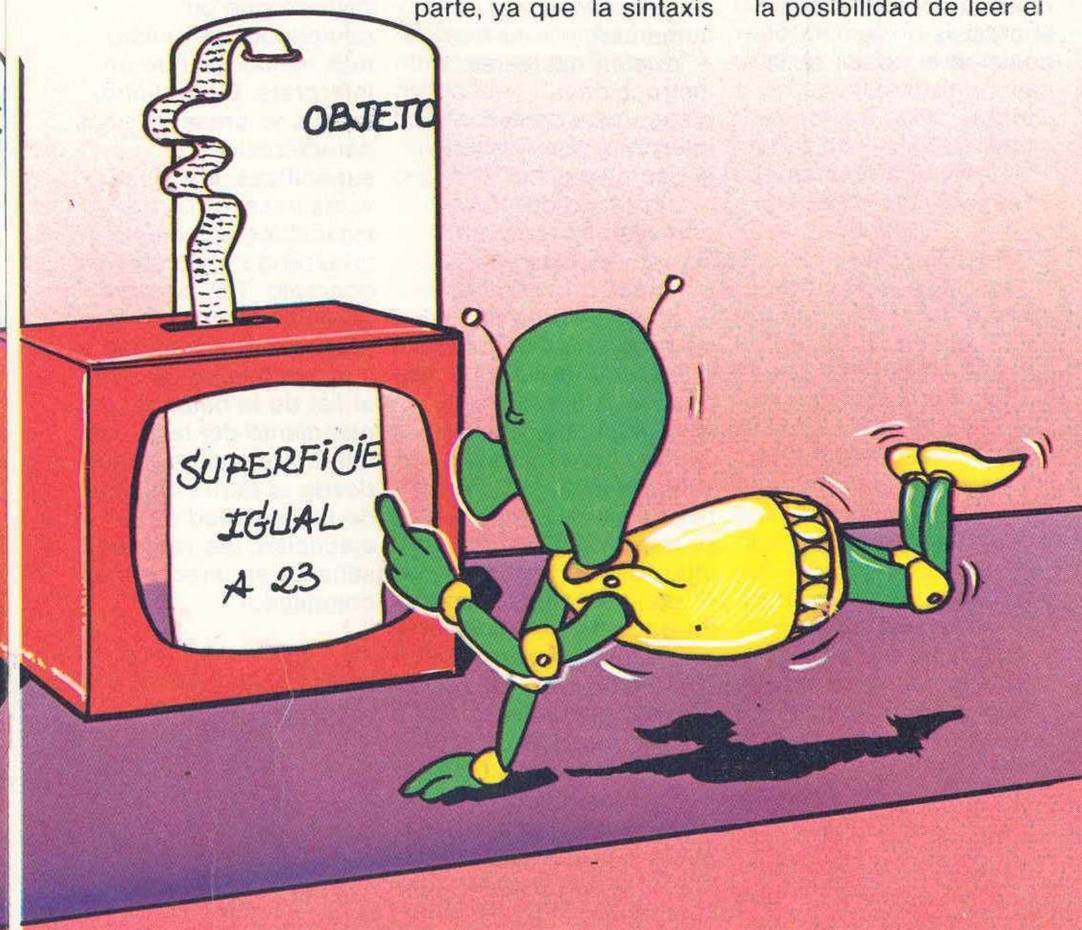
Se podría pensar, para aumentarla, en traducir manualmente el programa BASIC en el correspondiente programa escrito en código máquina, pero sería una operación larga y aburrida, que habría que repetir después con cada variación del programa.

Las operaciones mecánicas, largas, aburridas, y repetitivas, son precisamente el tipo de cosas que podemos hacer ejecutar a un ordenador: bastará, por tanto, escribir (una sola vez) un programa que

FUENTE

0100110
0011101
001110
000111

traduzca cada instrucción BASIC a una rutina correspondiente en código máquina, y nuestros «problemas» quedarán resueltos. Un programa de este tipo se llama compilador BASIC: lee nuestro texto y, con paciencia, escribe (o, mejor, compila) un programa en código máquina que hace el mismo trabajo. La primera ventaja de un programa compilado es la eliminación del procedimiento de interpretación de las instrucciones. Por otra parte, ya que la sintaxis se comprueba durante la compilación, se ahorra más tiempo, y, además, no existe el riesgo de que el programa contenga errores de sintaxis no detectados. Pero la mayor ventaja viene del hecho de que el compilador, al tener la posibilidad de leer el



texto integro BASIC, puede eliminar las lentas operaciones de búsqueda de las lineas (que el intérprete BASIC debe ejecutar a cada GOTO o GOSUB) y de las variables (para utilizar una variable el traductor ejecuta una serie de operaciones, referidas principalmente al análisis de la posición ocupada en la memoria por esta variable).

Por todos estos factores la velocidad de un programa compilado es notablemente superior a la del mismo programa interpretado.
Naturalmente, también la compilación tiene sus desventajas:
– las dimensiones del programa, generalmente, aumentan;
– existen bastantes instrucciones disponibles con el intérprete pero no con el compilador;
– el trabajo de compilación requiere tiempo (incluso varios minutos);
– no es posible detener un programa compilado y reemprenderlo después de, por ejemplo, haber impreso el valor de cualquier variable (algo perfectamente licito para el BASIC interpretado);
– cada modificación se introduce en el texto original, que debe ser compilado de nuevo.
Generalmente, se prefiere escribir un programa en BASIC normal (es decir, interpretado), ponerlo a punto y probarlo a fondo, y después compilarlo.
Un programa compilado ya no es BASIC, sino código máquina. En consecuencia no puede ser cargado en memoria con un simple LOAD, o puesto en marcha con un RUN, sino que es necesario usar otras instrucciones.
De lo que hemos dicho hasta ahora, se podria deducir que un compilador es mucho más ventajoso que un intérprete. En realidad, cada uno presenta unas caracteristicas especificas, algunas ventajosas, otras no, evaluables solamente analizando un problema concreto. Ten siempre presente que, en lo que respecta a la facilidad y a la flexibilidad de uso, el fiel de la balanza cae netamente del lado del intérprete, mientras que desde el punto de vista de la velocidad de la ejecución, las ventajas señalan en dirección al compilador.

Representación de los números

Hasta ahora hemos evitado siempre cualquier cosa que no haya sido una aritmética bastante simple, y continuaremos haciendo lo mismo. Pero sería un error no echar un breve vistazo a la capacidad aritmética del ordenador y, sobre todo, a la representación de números en el ordenador.
La elaboración de un dato requiere de hecho que este esté ya contenido en la memoria del ordenador, de forma que sea inmediatamente utilizable por la unidad central. Ya que la única representación que la CPU puede reconocer es la digital, también los números deben ser expresados de esta forma.
Ya conoces como se pasa de la notación decimal a la binaria y viceversa, así que no hablaremos aquí de este tema.
Examinaremos, sin embargo, algunos aspectos más interesantes que derivan del uso práctico de los números con el ordenador.
Antes de nada, el consejo más importante es examinar siempre con una cierta dosis de desconfianza cualquier número que haya sido utilizado en el interior de un programa. La razón de esta sugerencia es que los números sufren casi siempre en el interior del ordenador redondeos y truncamientos, debidos a la conversión de decimal a binario, que modifican –aunque sea ligeramente– su valor exacto. Por ejemplo, en nuestra familiar notación decimal, recordarás que un número como 1/3 o 1/7 no puede tener una expresión exacta: 1/3=0.333333...
Nosotros damos por descontado que añadiendo al número tantos 3 como queramos, podemos obtener un grado de aproximación aceptable para cualquier problema concreto. Del mismo modo, en el sistema binario algunos números no pueden ser

expresados exactamente: por ejemplo, en la forma decimal podemos decir que 1/10=0.1 pero cuando el número es transformado en binario no puede ser representado con la precisión debida.
Para comprobarlo en la práctica es suficiente que teclees y ejecutes este pequeño programa:

```
10 LET I=1
20 LET I=I-0.1
30 IF I=0 THEN STOP
40 PRINT I
50 GOTO 20
```

Teóricamente, todo es correcto. Estas instrucciones constituyen un bucle que debería restarle 10 veces la cantidad 0.1 al número 1. La línea 30 indica al ordenador que debe detenerse cuando la variable I valga 0.
En cambio, como habrás podido comprobar, la condición I=0 nunca se alcanza, y tu Spectrum seguirá restando a I el número 0.1 infinitamente. Los resultados de las diversas operaciones aparecerán poco a poco en la pantalla, mostrándote inmediatamente la causa de este comportamiento, es decir, la aproximación introducida en el resultado por los diversos cálculos.
En general, es siempre mejor desconfiar de las cifras menos significativas (es decir, la cifra del extremo derecho) de cualquier resultado numérico: podría ser consecuencia de imprecisiones internas.
Para que nuestro programa hubiera funcionado como nosotros queríamos, tendría que haber sido modificada esta línea:

```
30 IF I<0.1 THEN STOP
```

y todo habría funcionado correctamente.
Ya hemos visto que los números pueden sufrir alteraciones dentro de la memoria.
Existen en cualquier caso unos límites al número de cifras que un ordenador es capaz de tratar: números demasiado grandes o demasiado pequeños deben ser redondeados casi necesariamente.
Veamos mejor esto.
Un número muy grande (o quizá, muy pequeño) puede ser representado por la llamada notación científica, es decir, con la forma abreviada del propio número, en la cual se indica la potencia de 10 a la cual el número debe ser elevado, precedida de la letra «E» (en la práctica, esta significa cuántas cifras tiene que

desplazarse la coma). La parte a la izquierda de E se llama mantisa; la siguiente, exponente.

Forma normal	Notación científica
1000000	1E+06
0.00000001	1E–07
0.000586321	0.586321E-3
93000000000	0.93E+11

El ordenador, en el momento de la introducción de los números en forma normal, los transformará inmediatamente a este formato (llamado también de coma flotante), consistente en poner el punto decimal detrás de la primera cifra significativa y ajustar el exponente en consecuencia. Naturalmente, si las mantisa supera una cierta longitud, algunas cifras se perderán. Así, los dos números.

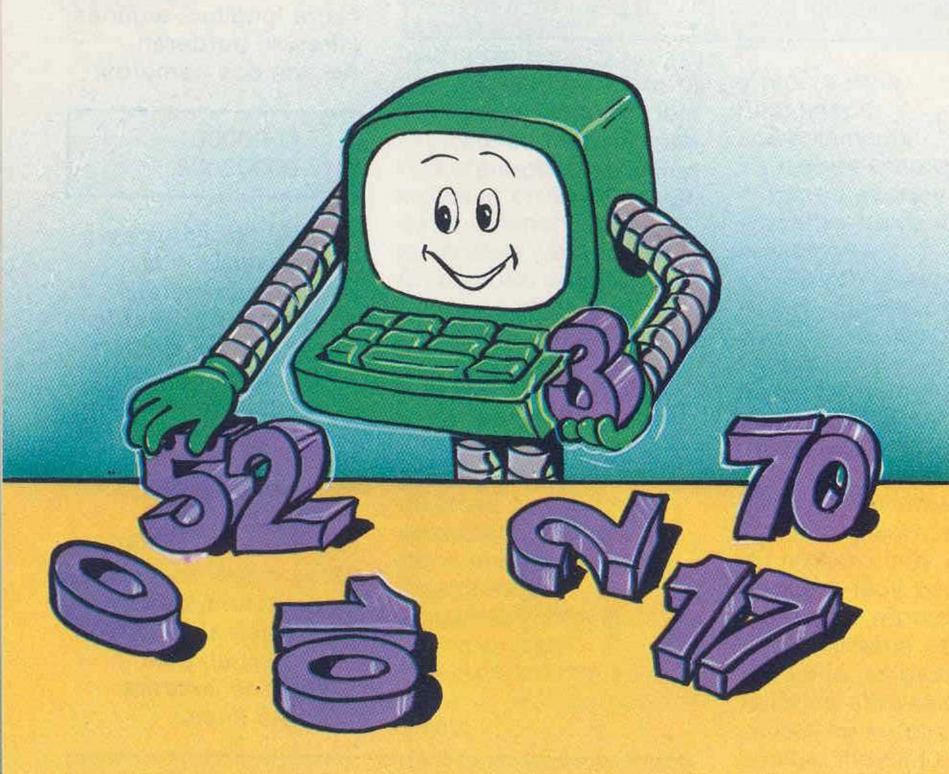
```
12.00000001
12.00000008
```

no son diferentes para tu ordenador, y la información menos significativa se pierde. Por supuesto, existen límites para el exponente. Cuando el BASIC imprime un número, lo hace en formato normal, al menos que esto comporte un número de cifras mayor que el de su precisión, en cuyo caso pasa al formato exponencial. Este ejemplo te ilustrará mejor lo dicho:

```
10 PRINT "NUMERO", "REPRESENTACION"
20 PRINT
30 FOR I=–10 TO 10
40 PRINT "10↑"; I, 10↑I
50 NEXT I
```

Funciones numéricas

Ya casi hemos terminado el examen de las diversas funciones disponibles en tu Spectrum: de hecho, solamente nos quedan dos. Dentro de poco veremos además como es posible, a través de un útil y potente instrumento proporcionado por el BASIC, crear y definir a placer otra función, que no forme parte del grupo del que se dotó al ordenador en la fábrica.



EXP

La función exponencial se utiliza para elevar el número «e» a una potencia determinada. Esta función calcula el valor de la constante e (base de los logaritmos llamados naturales) y lo eleva a la potencia proporcionada por el argumento especificado. El número e es un valor concreto, que se define matemáticamente de forma muy precisa, pero del cual tu Spectrum utiliza una aproximación equivalente a 2.7182818.
Veámosla trabajando:

```
PRINT EXP(1)
```

pide al ordenador que visualice el resultado de e elevado a 1. En la pantalla aparecerá el valor 2.7182818.
El origen de este valor es bastante complejo; este pequeño programa te ilustrará como este número puede ser obtenido de forma aproximada:

```
10 FOR I=1 TO 1000
20 LET E=(1+1/I)↑I
30 PRINT E, EXP(1)
40 NEXT I
```

A la izquierda de la pantalla aparecerá una serie de números, que con el aumento de I tenderá a ser siempre más próxima al valor de e proporcionado por la función EXP (visible a la derecha).

Sintaxis de la función

EXP (expresión)

LN

La función LN se usa para realizar la operación exactamente inversa a la ejecutada con EXP.
LN calcula el logaritmo natural, es decir en base e, del argumento. Los argumentos se intercambian con los resultados, y viceversa.
El logaritmo es una función particularmente útil y preciosa en numerosos cálculos matemáticos, ya que permite abreviar cuentas y operaciones, gracias a algunas importantes propiedades de las que goza.
Sobre todo, en cálculos técnicos y científicos resulta de importancia básica poder servirse de los logaritmos.
De todas formas, si no conoces sus reglas de utilización, no te preocupes: dado que no has tenido necesidad de ellos hasta ahora, quizá los logaritmos tampoco te sirvan para nada en el futuro. En el caso de que, de cualquier forma, desees usarlos (quizás para generar números raros o cosas por el

estilo), acuérdate de no asignarle jamás a LN un argumento negativo o nulo. Al igual que la función SQR, tampoco LN puede trabajar con números inferiores a cero (y tampoco con cero).

Sintaxis de la función

LN (expresión)

DEF FN

Ocurre a menudo que en el mismo programa estén presentes cálculos o expresiones parecidas; en este caso es posible escribirlos una sola vez, para después llamarlos mediante una subrutina. Pero estas últimas tienen el inconveniente de trabajar con los nombres de variables especificadas en la propia rutina.
Por ejemplo, si quisiéramos calcular el valor medio de dos números, podríamos escribir una cosa como ésta:

```
100 REM LA SUBRUTINA PROPORCIONA
110 REM LA MEDIA DE X E Y
120 LET MEDIA=(X+Y)/2
130 RETURN
```

Sin embargo, esta subrutina proporciona solamente la media de los valores contenidos en las variables X e Y. Podría suceder que en nuestro programa necesitáramos también la media de P y Q, o de C y D. Para usar la subrutina tendremos que reasignar los valores de las variables cada vez:

```
20 LET X=P : LET Y=Q : GOSUB 100
30 LET X=C : LET Y=D : GOSUB 100
```

Pero en todas las funciones presentes en el Spectrum se hace en cambio referencia a las variables interesadas, escribiendo así SQR(A) para obtener la raíz cuadrada de A y SQR(Z)



para pedir la de Z. Definiendo una nueva función, podremos igualmente especificar las variables sobre cuáles efectuar la operación.

El BASIC le permite al programador definir nuevas funciones matemáticas, utilizables a continuación como si formaran parte del lenguaje en sí. Para definir una nueva función es necesario insertar en el programa una instrucción del tipo

```
DEF FN nombre (argumento)=expresión
```

En el caso de nuestro ejemplo la solución habría podido ser:

```
10 DEF FN M(X,Y)=(X+Y)/2
```

De ahora en adelante la función será considerada como definida y el ordenador la ejecutará con toda tranquilidad. La instrucción DEF FN permite, por tanto, definir nuevas funciones, además de las que forman parte del propio lenguaje. Profundicemos en la sintaxis a utilizar:

– DEF FN es una palabra reservada, que indica al ordenador la definición de una nueva función;

– nombre es la palabra que se utilizará en el resto del programa para designar la función definida con DEF FN; debe ser una letra si el resultado de la función es un número, y una letra y un dólar cuando el resultado sea en cambio una cadena;

– argumentos son las variables que sirven para definir los operadores sobre los cuales intervendrá la función;

– expresión indica las operaciones que el ordenador deberá ejecutar para alcanzar el resultado.

Después de que el programa ha ejecutado la línea

```
DEF FN M(X, Y) = (X+
Y)/2
```

la función M(X,Y) se convierte en una palabra reservada del BASIC y puede ser usada como si formara parte del lenguaje. Por ejemplo:

```
300 A=FN M(C,S)
```

quiere decir: calcula el valor de A, aplicando al valor de C y de S la función M definida anteriormente.

La X y la Y, que habíamos usado en la línea 10 para definir la función, son sustituidas por el valor de la expresión que sigue a FN M, esto es, por el valor de C y S. Se dice entonces que X e Y son parámetros formales (o parámetros ficticios), puesto que su único objeto es indicar dónde van a ser empleados los argumentos de la función cuando ésta sea llamada.
Naturalmente, para que el BASIC pueda reconocerla, la definición de la función debe estar presente en el programa.
Aquí debajo puedes encontrar una serie de funciones matemáticas que no están comprendidas en el BASIC estándar, pero que es posible definir mediante DEF FN en el momento en el que te sean necesarias:

SEC(X)=1/COS(X)	Secante
CSC(X)=1/SIN(X)	Cosecante
COT(X)=1/TAN(X)	Cotangente
ARCSIN(X)=ATN(X/SQR(−X*X+1))	Arcoseno (Seno inverso)
ARCCOS(X)=−ANT(X/SQR(−X*X+1))+1.5708	Arcocoseno (Coseno inverso)
ARCSEN(X)=ATN(SQR(X*X−1))+(SGN(X)−1)*1.5708	Arcotangente (Tangente inversa)
ARCCSC(X)=ATN(1/SQR(X*X−1))+(SGN(X)−1)*1.5708	Cosecante inversa
ARCCOT(X)=−ATN(X)+1.5708	Cotangente inversa
SINH(X)=(EXP(X)−EXP(−X))/2	Seno hiperbólico
COSH(X)=(EXP(X)+EXP(−X))/2	Coseno hiperbólico
TANH(X)=−EXP(−X)/(EXP(X)+EXP(−X))*2+1	Secante hiperbólica
SECH(X)=2/(EXP(X)+EXP(−X))	Tangente hiperbólica
CSCH(X)=2/(EXP(X)−EXP(−X))	Cosecante hiperbólica
COTH(X)+EXP(−X)/EXP(X)−EXP(−X))*2+1	Cotangente hiperbólica
ARGSINH(X)=LN(X+SQR(X*X+1))	Seno hiperbólico inverso
ARGCOSH(X)=LN(X+SQR(X*X−1))	Coseno hiperbólico inverso
ARGTANH(X)=LN((1+X)/(1−X))/2	Tangente hiperbólica inversa
ARGSECH(X)=LN((SQR(−X*X+1)+1)	Secante hiperbólica inversa
ARGCSCH(X)=LN(SGN(X)*SQR(X*X+1)+1)/X	Cosecante hiperbólica inversa
ARGCOTH(X)=LN((X+1)/(X−1))/2	Cotangente hiperbólica inversa
MOD(A)=INT((A/B−INT(A/B))*B+0.5)*SGN(A/B)	Módulo
LN(X)=LN(X)/LN(B)	Logaritmo en base B

Ejemplos

```
10 DEF FN A(R)=R*R*PI
20 LET X=2
30 PRINT FN A(X)
```

Define una función (cuyo nombre es A) que tiene como objeto calcular el área del círculo de radio R. R es el parámetro formal. La línea 20 calcula el área del círculo de radio 2. X constituye en cambio el parámetro real.

```
10 DEF FN S(X)=SIN(X)
```

Define una nueva función, llamada S, que ejecuta la misma operación que la propia función SIN.

```
40 PRINT FN S(3)
50 PRINT SIN(3)
```

El efecto de estas instrucciones será, por lo tanto, la impresión de dos valores absolutamente idénticos.

```
90 DEF FN A(X)=PEEK(2040)+PEEK(2041)*256
```

Esta función proporciona como resultado un valor independiente del argumento utilizado. Es perfectamente lícito (siempre que se respeten las reglas que impone la instrucción) definir funciones que, aunque lo pidan, no utilicen el argumento en el interior de la expresión. En este caso es también posible no indicar ningún argumento, dejando los paréntesis vacíos.

Habriamos podido escribir también DEF FN A()=PEEK(2040)+ PEEK(2041)*256 y el resultado no seria distinto.

```
20 DEF FN S(X)=3+SIN(2.3)
30 PRINT FN S(41)
40 PRINT FN S(3.82)
```

Tampoco en este caso el argumento tendrá ninguna influencia en el resultado, ya que no aparece en la expresión. Las dos instrucciones de impresión, a pesar de llamar a funciones con diferente argumento, producirán resultados idénticos.

```
DEF FN L$(A$,C)=A$(1 TO C)
```

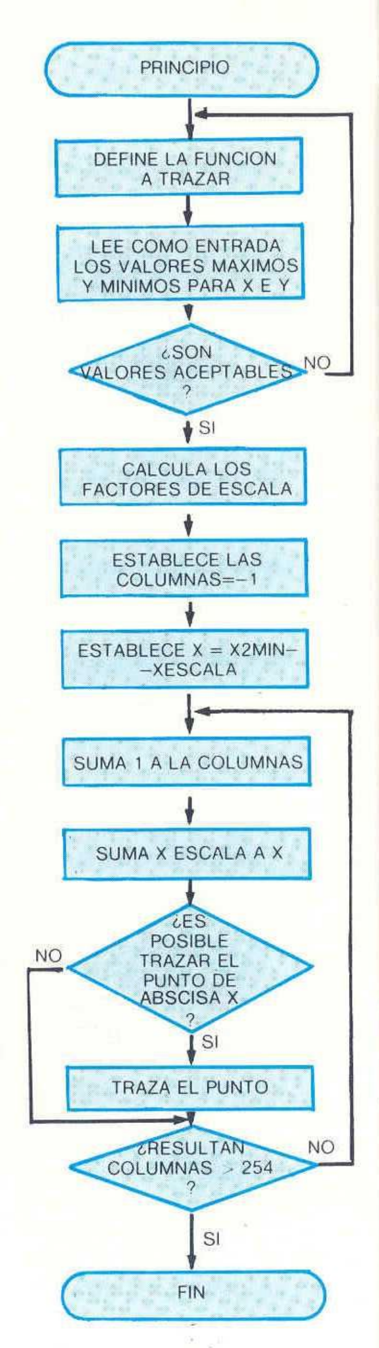
Define una función que extrae de una cadena A$ los primeros C caracteres. Naturalmente, ya que se trata de una función de tipo cadena, es preciso incluir el dólar.

Sintaxis de la función

DEF FN nombre función (variable [, variable]) = expresión

Dibujo de un gráfico

Ahora que ya sabemos definir a nuestro gusto todas las funciones que queramos, es posible escribir un programa para ejecutar una útil y divertida operación: el trazado del gráfico de una función.
Para alcanzar este objetivo deberemos, naturalmente, retomar las instrucciones elementales que habíamos presentado algunas lecciones atrás, hablando de los gráficos, y que servirán para trazar los puntos concretos que compondrán la función.
Desde el punto de vista conceptual la escritura del programa no es muy difícil: basta con definir la función que deseamos dibujar y proceder a su trazado punto a punto.
El uso del programa resulta, en cambio, un poco más complicado: es necesario prestar siempre atención a las funciones que queremos dibujar.
Ocurre a menudo que algunas funciones no estén definidas (es decir, que no existan), en correspondencia con determinados valores del argumento. Por ejemplo, la función SQR no admite argumentos negativos (¡no existe la raíz cuadrada de un número negativo!) y cualquier intento de ejecutar cálculos con números menores que cero encontrará inmediatamente la barrera del mensaje de error.
Esta eventual limitación es muy fácil de evitar en el caso de funciones sencillas (como SQR), pero resulta extremadamente difícil cuando la propia función se complica (¿quién es capaz de adivinar rápidamente donde no está definida LOG(SIN(TAN(X))*SQR(COS(X)))?).
Dado que la única consecuencia de una situación de este tipo es como máximo la visualización de un mensaje de error, se podrá siempre (aunque no es la solución más elegante) proceder por tanteos sucesivos.
He aquí el diagrama de bloques del programa:



Y he aquí el correspondiente listado BASIC:

```
   5 DEF FN A(X)=SIN(X)
  10 CLS
  20 PRINT:PRINT "GRAFICO FUNCION"
  30 INPUT "VALOR MAXIMO DE LA X";X1MAX
  40 INPUT "VALOR MINIMO DE LA X";X2MIN
  50 INPUT "VALOR MAXIMO DE LA Y";Y1MAX
  60 INPUT "VALOR MINIMO DE LA Y";Y2MIN
  70 IF X1MAX<=X2MIN OR Y1MAX<=Y2MIN THEN RUN
  80 GOSUB 1000:REM FACTORES DE ESCALA
  90 LET ESCALA=176/(Y1MAX-Y2MIN)
 100 LET X=X2MIN-XESCALA
 110 LET C=-1
 120 REM
 130 REM TRAZA LA FUNCION
 140 REM
 150 LET C=C+1
 160 LET X=X+ESCALA
 170 LET Y=SK+(FN A(X)*ESCALA)
 180 IF Y>175 OR Y<0 THEN GOTO 200
 190 PLOT C,Y
 200 IF C<255 THEN GOTO 150
 210 REM
 220 REM GRAFICO EJECUTADO
 230 REM
 240 GO TO 240
1000 REM
1010 REM CALCULA LOS FACTORES DE ESCALA
1020 REM
1030 LET XESCALA=ABS((X1MAX-X2MIN)/256)
1040 LET YESCALA=ABS((Y1MAX-Y2MIN)/176)
1050 IF Y2MIN>=0 THEN LET SK=0:RETURN
1060 IF Y1MAX<=0 THEN LET SK=175:RETURN
1070 LET C=-1
1080 LET Y=Y2MIN-YESCALA
1090 LET C=C+1
1100 LET Y=Y+YESCALA
1110 IF Y>=0 THEN LET SK=C:RETURN
1120 GOTO 1090
```

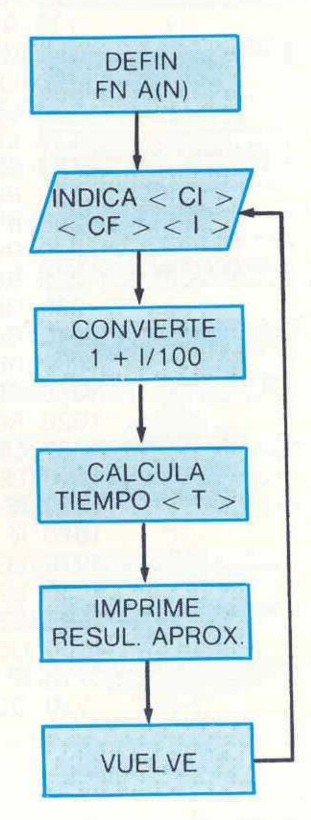
No existen especiales dificultades: lo único digno de comentario es el cálculo de los factores de escala, necesario para adecuar el gráfico a las dimensiones de nuestra cuadricula imaginaria, representada por nuestra pantalla. Estos factores dependerán de los factores máximos y mínimos que asignes a la X y a la Y (es decir, a la abscisa y a la ordenada).
Los abudantes comentarios y el uso frecuente de subrutinas hacen que el programa resulte bastante sencillo de leer y comprender.
Para cambiar la función a trazar, la única modificación necesaria es la de sustituir en la línea 5 la expresión de la antigua función (en la actualidad SIN(X)) por la definición de la nueva.
A título de posibles mejoras podrías intentar como ejercicio trazar antes y después del verdadero gráfico también los dos ejes cartesianos tomados como sistema de referencia (¡ten cuidado porque habrá que tener en cuenta varias posibilidades!).

Programa financiero

El programa siguiente emplea la fórmula:

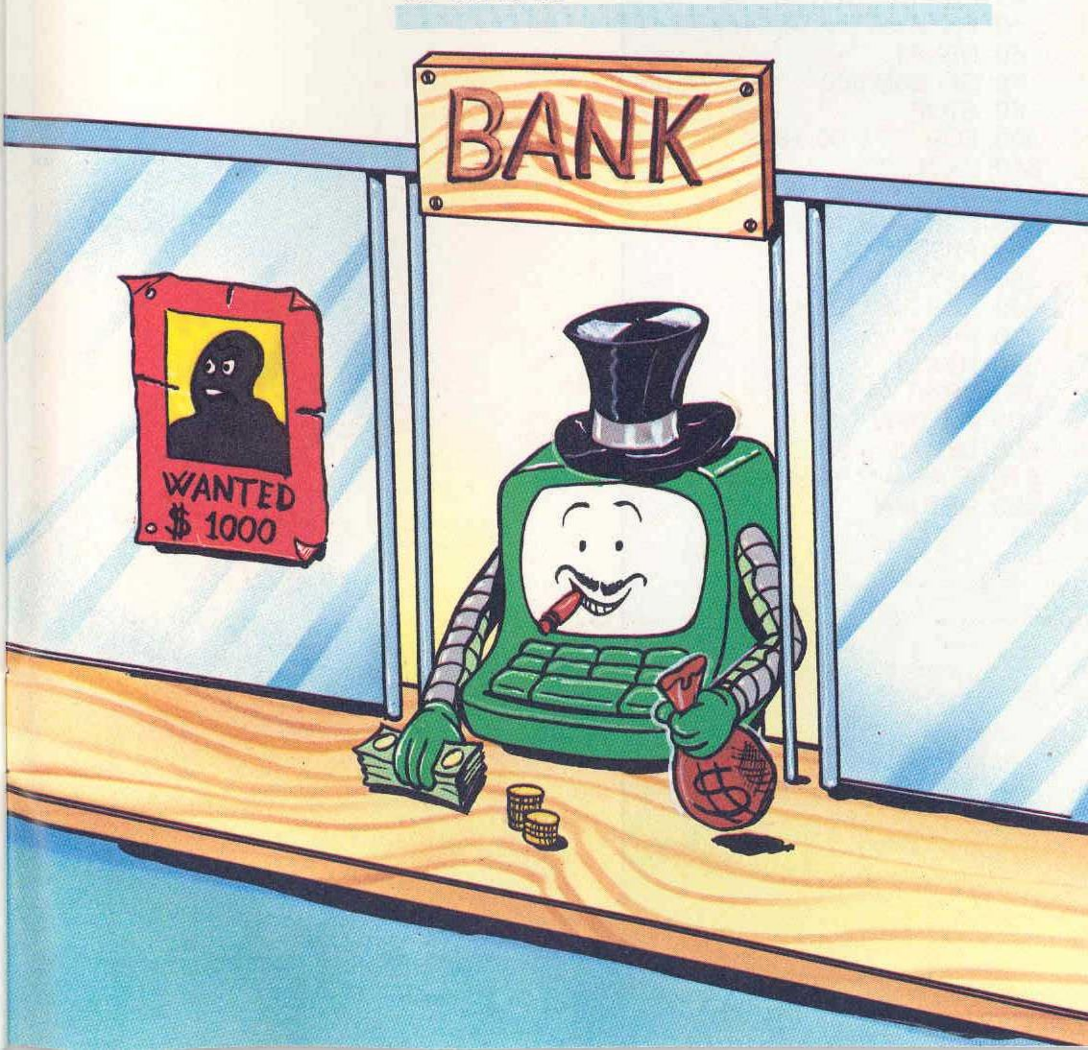
T=LN (CF/C)/LN (I)

para calcular el tiempo necesario para que un determinado capital inicial (CI) llegue a convertirse en el capital final (CF) con un



determinado tipo de interés (I).
Observa la función definida en la primera línea, cuya tarea es la de efectuar el redondeo del tiempo calculado.

```
10 DEF FN A(N) = INT (N + 5)
20 INPUT "CI ="; CI' "CF ="; CF' "I ="; I
30 LET I = 1 + I/100
40 LET T = LN (CF/CI)/LN I
50 PRINT FN A (T)
60 GOTO 20
```

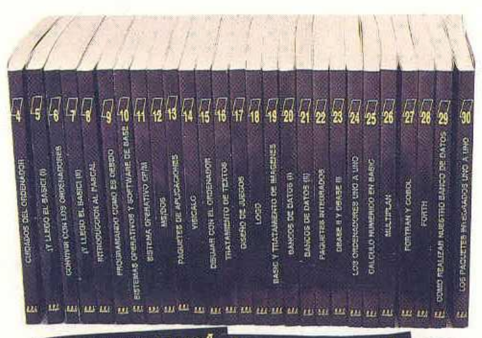


EJERCICIOS

La función definida en el programa siguiente realiza los cálculos necesarios para visualizar una parábola. Define otra distinta para la representación de otro gráfico.

```
 10 CLS
 20 GO SUB 300
 30 DEF FN X (Y) = Y * Y/2.6
 35 PRINT "PARABOLA"
 40 FOR I = –9 TO 9
 50 PRINT AT I + 10, FN X (I); "*"
 60 NEXT I
 70 GO SUB 500
 80 STOP
300 FOR I = 1 TO 10
310 PRINT "I"
320 NEXT I
330 FOR I = 1 TO 32
340 PRINT "–";
350 NEXT I
360 FOR I = 1 TO 10
370 PRINT "I"
380 NEXT I
390 PRINT AT 0, 0;
400 RETURN
500 LET A$ = INKEY$
510 IF A$ = "" THEN GO TO 500
520 RETURN
```

UNA GRAN OBRA A SU ALCANCE



UNA OBRA COMPLETISIMA EN 30 VOLUMENES QUE TRATA TODOS LOS TEMAS , DESDE QUE ES UN ORDENADOR HASTA EL ESTUDIO DE LOS DIVERSOS LENGUAJES, PASANDO POR LOS LENGUAJES, METODOS DE PROGRAMACION, ELECCION DEL ORDENADOR ADECUADO, DICCIONARIO, ETC.



B.B.I.
INGELEK

30 EXTRAORDINARIOS VOLUMENES DE APARICION SEMANAL CON TODOS LOS CONCEPTOS DE LA INFORMATICA

GRAN OFERTA DE SUSCRIPCION
9.995 PTAS

AHORRE MAS DE 1.000 PTAS Y LLEVESE UNA MAGNIFICA CALCULADORA SOLAR VALORADA EN 2.500 PTAS.



OFERTA VALIDA UNICAMENTE PARA ESPAÑA

SUSCRIBASE POR TELEFONO

Todos los días, excepto sábados y festivos, de 8 a 6,30 atenderemos sus consultas en el



2505820